1. Implementasikan fitur CRUD untuk data kontak dengan menggunakan SQLite.

* DatabaseSQlite.dart, untuk dapat menggunakan SQLite saya menambahkan package sqflite di pubspec.yaml

import 'package:sqflite/sqflite.dart' as sql;

class DatabaseSQlite {

static Future<void> createTables(sql.Database database) async {

await database.execute('''CREATE TABLE contacts (

id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

nama TEXT,

no\_telp VARCHAR,

tanggal VARCHAR,

color VARCHAR,

file VARCHAR

)

''');

}

static Future<sql.Database> db() async {

return sql.openDatabase('contactsnew.db', version: 1,

onCreate: (sql.Database database, int version) async {

await createTables(database);

});

}

// tambah

static Future<int> tambahContact(String nama, String no\_telp, String date,

String color, String file) async {

final db = await DatabaseSQlite.db();

final data = {

'nama': nama,

'no\_telp': no\_telp,

'tanggal': date,

'color': color,

'file': file

};

return await db.insert('contacts', data);

}

// ambil data

static Future<List<Map<String, dynamic>>> getContacts() async {

final db = await DatabaseSQlite.db();

return db.query('contacts');

}

//ubah data

static Future<int> ubahContacts(int id, String nama, String no\_telp,

String date, String color, String file) async {

final db = await DatabaseSQlite.db();

final data = {

'nama': nama,

'no\_telp': no\_telp,

'tanggal': date,

'color': color,

'file': file

};

return await db.update('contacts', data, where: "id = $id");

}

//hapus data

static Future<int> hapusContacts(int id) async {

final db = await DatabaseSQlite.db();

// final data = {'id': id, 'nama': nama, 'no\_telp': no\_telp};

return await db.delete('contacts', where: "id = $id");

}

}

* Contacts.dart

@override

void initState() {

super.initState();

refreshContacts();

checkLoginStatus();

}

//fungsi refresh contacts

void refreshContacts() async {

final data = await DatabaseSQlite.getContacts();

setState(() {

contacts = data;

});

}

//fungsi tambah contacts

Future<void> tambahContact() async {

await DatabaseSQlite.tambahContact(

namaController.text,

nomorController.text,

\_selectedDate.toString(),

pickerColor.toString(),

\_filePath.toString());

refreshContacts();

}

//fungsi ubah contacts

Future<void> ubahContacts(int id) async {

await DatabaseSQlite.ubahContacts(

id,

namaController.text,

nomorController.text,

\_selectedDate.toString(),

pickerColor.toString(),

\_filePath.toString());

refreshContacts();

}

void formContacts(int id) async {

final dataContacts = contacts.firstWhere((element) => element['id'] == id);

namaController.text = dataContacts['nama'];

nomorController.text = dataContacts['no\_telp'];

\_selectedDate = DateTime.parse(dataContacts['tanggal']);

pickerColor = Color(int.parse(

dataContacts['color'].toString().split('(')[1].split(')')[0]));

\_filePath = dataContacts['file'];

}

//fungsi hapus contacts

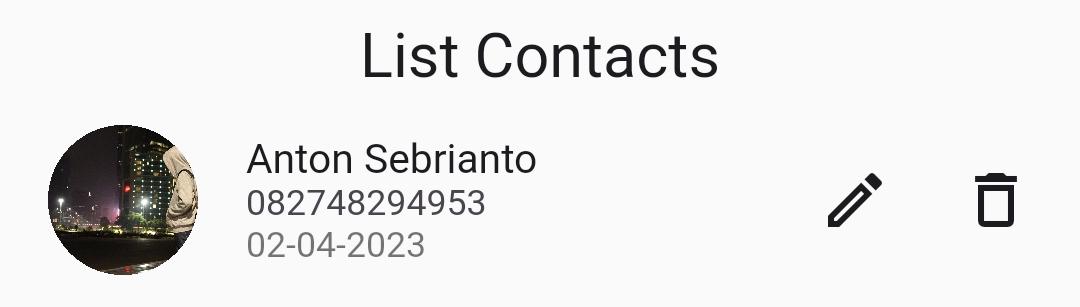
Future<void> hapusContacts(int id) async {

await DatabaseSQlite.hapusContacts(id);

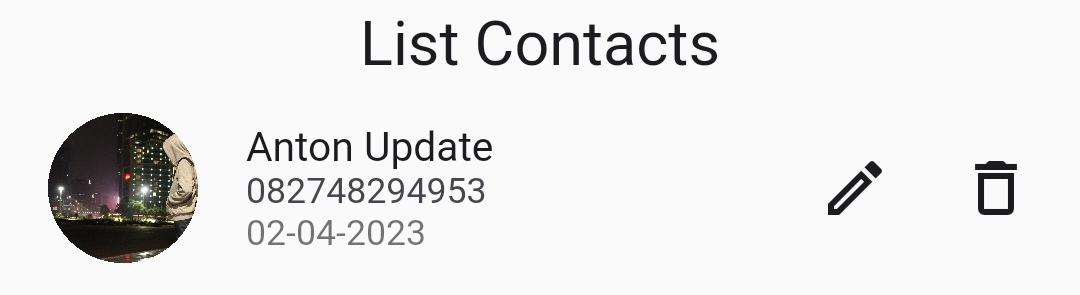
refreshContacts();

}

* Create contacts menggunakan SQlite berhasil dan berhasil melakukan read contacts menggunakan SQlite



* Update contacts menggunakan SQlite berhasil. Untuk mengupdate atau mengedit contacts menggunakan icon edit yang ada di samping dari nama contacts



* Delete contacts menggunakan SQlite berhasil. Untuk menghapus atau delete contacts menggunakan icon delete yang ada di samping icon edit.

